#8

IN THE MATTER OF
Patent Application of
SONY COMPUTER ENTERTAINMENT INC.

I, Takayuki ICHIKAWA, of 25-16, Motomachi, Tokorozawa-shi, Saitama 359-1121, Japan, do hereby declare that I am conversant with the Japanese and English languages and am a competent translator thereof. I further declare that to the best of my knowledge and belief, the following is a true and correct translation, made by me, of the official copy of the document in respect of a Patent Application No. 11-257933 filed in Japan on September 10, 1999.

Signed this 12th day of September, 2003.

By Takayuki Ichikawa
Takayuki ICHIKAWA

(Translation)

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: September 10, 1999

Application Number: Patent Application No. 11-257933

Applicant(s): SONY COMPUTER ENTERTAINMENT INC.

June 29, 2000

Commissioner,
Patent Office

Takahiko KONDO (seal)

(Certificate No. 2000-3050124)

[Document Name] Patent Application

[Reference Number] SCEI99019

[Date of Submission] September 10, 1999

[To] The Commissioner of the Patent Office

[International Patent Classification] G06T 15/00

[TITLE OF THE INVENTION] METHOD OF AND APPARATUS FOR RENDERING IMAGE AND RECORDING MEDIUM

[Number of claim(s)] 15

[Inventor]
[Domicile or Residence] c/o SONY COMPUTER ENTERTAINMENT INC.
1-1, Akasaka 7-chome, Minato-ku, Tokyo
[Name] Shinya MATSUMOTO

[Inventor]
[Domicile or Residence] c/o SONY COMPUTER ENTERTAINMENT INC.
1-1, Akasaka 7-chome, Minato-ku, Tokyo
[Name] Takuya SEKI

[Inventor]
[Domicile or Residence] c/o SONY COMPUTER ENTERTAINMENT INC.
1-1, Akasaka 7-chome, Minato-ku, Tokyo
[Name] Yuta KIMURA

[Inventor]
[Domicile or Residence] c/o SONY COMPUTER ENTERTAINMENT INC.
1-1, Akasaka 7-chome, Minato-ku, Tokyo
[Name] Yasuhiro SHIMURA

[Applicant]
[Identification Number] 395015319
[Name] SONY COMPUTER ENTERTAINMENT INC.

[Agent]
[Identification Number] 100077665
[Patent Attorney]
[Name] Yoshihiro CHIBA

[Selected Agent]
[Identification Number] 100096518
[Patent Attorney]
[Name] Hiroshi TUCHIYA

[Indication of Fees]
[Prepayment Register Number] 001834
[Amount of Payment] 21000

[List of the Attached Documents]
[Document Name] Specification 1
[Document Name] Drawings 1
[Document Name] Abstract 1
[Power of Attorney Number] 9908317
[Proof] necessary

[DOCUMENT NAME] Specification

[TITLE OF THE INVENTION]

METHOD OF AND APPARATUS FOR RENDERING IMAGE AND RECORDING MEDIUM

[CLAIMS]

[Claim 1]

A method of rendering an image, comprising the step of:

moving a texture simulatively on at least one object thereby to render an image.

[Claim 2]

A method according to claim 1, wherein said moving step comprises the steps of:

placing one or more semitransparent textures on the surface of at least one semitransparent or transparent object; and

moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 3]

A method according to claim 2, wherein said moving step further comprises the step of arranging said semitransparent or transparent object in one or multiple layers.

[Claim 4]

A method according to claim 1, wherein said moving step comprises the steps of:

placing one or more semitransparent textures on the surface of a plurality of semitransparent or transparent polygons which make up said object; and

moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 5]

A method according to claim 4, wherein said moving step further comprises the step of arranging said one or more semitransparent or transparent polygons in one or multiple layers.

[Claim 6]

An apparatus for rendering an image, comprising:

rendering means for moving a texture simulatively on at least one object thereby to render an image.

[Claim 7]

An apparatus according to claim 6, wherein said rendering means comprises:

means for placing one or more semitransparent textures on the surface of at least one semitransparent or transparent object, and moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 8]

An apparatus according to claim 7, wherein said rendering means comprises:

object setting means for arranging said semitransparent or transparent object in one or multiple layers.

[Claim 9]

An apparatus according to claim 6, wherein said rendering means comprises:

means for placing one or more semitransparent textures on the surface of a plurality of semitransparent or transparent polygons which make up said object, and moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 10]

An apparatus according to claim 9, wherein said rendering means comprises:

object setting means for arranging one or more semitransparent or transparent polygons in one or multiple layers.

[Claim 11]

A recording medium storing a program and data, said program comprising the step of:

moving a texture simulatively on at least one object thereby to render an image.

[Claim 12]

A recording medium according to claim 11, wherein said moving step comprises the steps of:

placing one or more semitransparent textures on the surface of at least one semitransparent or transparent object; and

moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 13]

A recording medium according to claim 12, wherein said moving step further comprises the step of:

arranging said semitransparent or transparent object in one or multiple layers.

[Claim 14]

A recording medium according to claim 11, wherein said moving step comprises the steps of:

placing one or more semitransparent textures on the surface of a plurality of semitransparent or transparent polygons which make up said object; and

moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

[Claim 15]

A recording medium according to claim 14, wherein said moving step further comprises the step of:

arranging said semitransparent of transparent polygons which make up said object in one or multiple layers.

[DETAILED EXPLANATION OF THE INVENTION]

[0001]

[TECHNICAL FIELD TO WHICH THE INVENTION PERTAINS]

The present invention relates to a method of and an apparatus for rendering an image representing a fluid motion

such as a stream of water or a flow of smoke, and a recording medium which stores a program and data for carrying out such an image rendering process.

[0002]

[PRIOR ART]

Recently, various computer graphics (CG) processing techniques including hidden line processing, hidden surface removal, smooth shading, texture mapping, etc. have been in rapid progress in combination with quickly growing hardware technologies.

[0003]

According to a general CG processing scheme, a plurality of three-dimensional shapes (objects) are generated by three-dimensional modeling of CAD, and a rendering process is performed by applying colors and shades to the objects, adding optical properties including mirror reflection, diffuse reflection, refraction, transparency, etc. to the objects, adding surface patterns to the objects, and plotting images depending on surroundings (such as window and scenery reflections and ambient light).

[0004]

[TASK TO BE SOLVED BY THE INVENTION]

For rendering a fluid motion such as a stream of water or a flow of smoke, for example, it has been attempted to move a number of objects at random and thereafter map an image of water or smoke onto the objects.

[0005]

Specifically, for processing ordinary polygons, a geometry operation unit performs perspective projection on the vertexes of polygons, and a rendering processor stores the obtained polygons in a display rendering memory.

[0006]

For example, if a fluid motion is to be rendered by moving n objects at random and then mapping a texture onto the objects, then it is necessary to perform the above process, (i.e., the ordinary polygon processing sequence with respect to an object) n times, and, as a result, it is necessary to perform (n × the number of polygon vertexes) perspective projection processes, and also to perform (n × the number of polygons) rendering processes.

[0007]

In order to render a stream of water or a flow of smoke realistically, therefore, a number of objects need to be prepared, and the period of time required to compute and render those objects increases in proportion to the number of objects involved. Consequently, those objects are displayed on a display monitor at a reduced speed.

[0008]

It is therefore an object of the present invention to provide a method of and an apparatus for rendering an image by moving a texture simulatively on at least one object thereby to greatly reduce the period of time required to calculate and render a fluid motion and also to render a high-quality image, and a recording medium which stores a

program and data for carrying out such an image rendering process.

[0009]

[SOLUTION FOR THE TASK]

According to the present invention, a texture is moved simulatively on at least one object thereby to render an image.

[0010]

Specifically, one or more semitransparent textures are placed on the surface of at least one semitransparent or transparent object, and the one or more semitransparent textures are moved simulatively in an arbitrary direction, thereby rendering an image. The semitransparent or transparent object may be arranged in one or multiple layers.

[0011]

Texture images that are moved simulatively on the surface of the object are superposed on the object, thereby generating (or animating) an infinite number of random patterns to express a fluid motion such as a stream of water or a flow of smoke.

[0012]

Alternatively, one or more semitransparent textures may be placed on the surface of a plurality of semitransparent or transparent polygons which make up the object, and the one or more semitransparent textures may be moved simulatively in an arbitrary direction, thereby rendering an

image. The semitransparent or transparent polygons may be arranged in one or multiple layers.

[0013]

Texture images that are moved simulatively on the surface of the polygons are superposed on the polygons, thereby generating or animating an infinite number of random patterns to express a fluid motion such as a stream of water or a flow of smoke.

[0014]

Since an image is rendered by moving a texture simulatively on at least one object, the period of time required to calculate and render a fluid motion can greatly be reduced, and a high-quality image can be rendered.

[0015]

[MODE FOR CARRYING OUT THE INVENTION]

An embodiment in which a method of and an apparatus for rendering an image according to the present invention are applied to an entertainment apparatus for performing three-dimensional CG processing (referred to as "an entertainment apparatus according to the present invention"), and a recording medium according to the present invention are applied to a recording medium storing a program and data executed by the entertainment apparatus, will be described below with reference to FIGS. 1 through 18.

[0016]

As shown in FIG. 1, an entertainment apparatus 10 according to the present invention comprises a main CPU 12

for controlling the entertainment apparatus 10, a main memory 14 for storing various programs to be run and various data, an image processor 18 for generating image data under the control of the main CPU 12 and outputting the generated image data to a display monitor 16, e.g., a CRT, and an input/output port 20 for sending data to and receiving data from external devices.

[0017]

The main memory 14, the image processor 18, and the input/output port 20 are connected to the main CPU 12 via a bus 22. To the input/output port 20, there are connected an input device 24 for inputting data (key entry data, coordinate data, etc.) to the entertainment apparatus 10, and an optical disk drive 26 for playing back an optical disk such as a CD-ROM or the like in which various programs and data (object-related data, texture data, etc.) are stored.

[0018]

The image processor 18 comprises a rendering engine 30, a memory interface 32, an image memory 34, and a display controller 36 such as a programmable CRT controller or the like.

[0019]

The rendering engine 30 serves to render image and store image data in the image memory 34 via the memory interface 32 based on a rendering command supplied from the main CPU 12.

[0020]

A first bus 38 is connected between the memory interface 32 and the rendering engine 30, and a second bus 40 is connected between the memory interface 32 and the image memory 34. Each of the first and second buses 38, 40 has a 128-bit width, for example, for allowing the rendering engine 30 to render and store image data in the image memory 34 at a high speed.

[0021]

The rendering engine 30 is capable of rendering image data of 320 × 240 pixels or image data of 640 × 480 pixels according to the NTSC or PAL system on a real-time fashion, i.e., more than ten times to several ten times in 1/60 seconds to 1/30 seconds.

[0022]

The image memory 34 is of a unified memory structure that is able to designate a texture rendering area 34a and a display rendering area 34b as the same area.

[0023]

The image controller 36 writes texture data read from the optical disk via the optical disk drive 26 or texture data generated in the main memory 14 via the memory interface 32 into the texture rendering area 34a of the image memory 34, and reads image data stored in the display rendering area 34b of the image memory 34 via the memory interface 32 and outputs the read image data to the display monitor 16 to display an image on its display screen.

[0024]

A characteristic function of the entertainment apparatus 10 will be described below with reference to FIGS. 2 through 18.

[0025]

According to the characteristic function, an image is rendered by moving a texture simulatively on at least one object.

[0026]

Specifically, as shown in FIG. 2, it is assumed that the water of a river 200 hits a post 202 standing in the river 200 and flows in divided streams on both sides of the post 202.

[0027]

As shown in FIG. 3, an object 204 representing the river 200 except for the post 202 is constructed of a number of semitransparent or transparent polygons 206, and a semitransparent texture image 208 (see FIG. 4A, for example) is mapped onto each of the polygons 206. The semitransparent texture image 208 may be an image expressing diffused reflections of light from waves, such as images containing randomly scattered semitransparent spots, as shown in FIGS. 4A and 4B.

[0028]

The texture image 208 to be used is stored in the texture rendering area 34a of the image memory 34. Thereafter, required polygons 206 are stored in the display

rendering area 34b of the image memory 34 and the texture image 208 is mapped onto each of the polygons 206, after which the polygons 206 with the mapped texture image 208 are displayed.

[0029]

Then, the texture image 208 in the texture rendering area 34a is moved a predetermined interval in an arbitrary direction and restored in the texture rendering area 34a, and the restored texture image 208 is mapped onto each of the polygons 206, after which the polygons 206 with the mapped texture image 208 are displayed.

[0030]

For example, an arbitrary texture image 208 is stored in the texture rendering area 34a in a pattern corresponding to the shape of the object 204 that is made up of the polygons 206.

[0031]

Specifically, texture images 208 covering five polygons 206 in a horizontal succession indicated by ① in FIG. 3 are stored in an area ① in the texture rendering area 34a as shown in FIG. 5, and texture images 208 covering five polygons 206 in a horizontal succession indicated by ② in FIG. 3 are stored in an area ② in the texture rendering area 34a as shown in FIG. 5.

[0032]

Then, a texture image 208 corresponding to one polygon 206 indicated by ③ in FIG. 3 is stored in an area ③ in the

texture rendering area 34a as shown in FIG. 5, and texture images 208 covering three polygons 206 in a horizontal succession indicated by ④ in FIG. 3 are stored in an area ④ in the texture rendering area 34a as shown in FIG. 5. Similarly, texture images 208 covering remaining polygons indicated by ⑤ - ⑧ in FIG. 3 are stored in corresponding areas in the texture rendering area 34a as shown in FIG. 5.

[0033]

Then, a texture image 208 indicated by (1) in FIG. 5 is mapped onto a polygon 206 indicated by (1) in FIG. 3. Likewise, other texture images 208 stored in the texture rendering area 34a are mapped onto corresponding polygons 206. When all the stored texture images 208 are mapped onto the corresponding polygons 206, an initial image is displayed.

[0034]

Thereafter, as shown in FIG. 5, each of the texture images 208 is moved a predetermined interval in the direction indicated by the arrow, and restored. Some texture images 208 (e.g., those indicated by ①, ②) are cycled to express an image in a circulating fashion. Specifically, when a succession of texture images 208 are moved a given interval, a texture image 208 is pushed out of the end of the succession and put back to the beginning of the succession.

[0035]

Especially, each of the texture images 208 indicated by

(6), (12), (16), (19) in FIG. 5 is moved in the two directions indicated by the arrows, and restored.

[0036]

In this manner, as shown in FIG. 2, it is possible to display a realistic image of the flowing water of the river 200, showing the water of the river 200 flowing in one direction and changing its direction around the post 202.

[0037]

In the above example, the characteristic function of the present invention is applied to a single object 204. However, the characteristic function of the present invention is also applicable to a number of objects arranged in superposed layers.

[0038]

For example, as shown in FIG. 6, two objects, i.e., first and second objects, 204a, 204b are prepared for the river 200 with the post 202 standing therein. Each of the first and second objects 204a, 204b is made up of a number of polygons 206. A first texture image 208a is mapped onto each of the polygons 206 of the first object 204a, and a second texture image 208b is mapped onto each of the polygons 206 of the second object 204b.

[0039]

When the first and second objects 204a, 204b are arranged in superposed layers, the first texture image 208a and the second texture image 208b are superposed, producing various patterns, as shown in an encircled area A.

[0040]

The first and second texture images 208a, 208b are moved on the first and second objects 204a, 204b with time in an arbitrary direction and restored in the texture rendering area 34a. Therefore, an infinite number of random patterns are generated or animated on the first and second objects 204a, 204b superposed in layers, making it possible to express a fluid motion such as a stream of water or a flow of smoke in a realistic manner.

[0041]

As shown in FIG. 7A, one example of a number of objects arranged in superposed layers is a scene in which water pouring from a faucet 210 is stored in a transparent water tank 212. In this example, as shown in FIG. 7B, an object 214 represents the manner in which water flows out from the faucet 210, two objects 216, 218 represent the manner in which water from the faucet 210 spreads out on the water surface in the water tank 212, and two objects 220, 222 represent the manner in which water is stored in the water tank 212.

[0042]

Each of the objects 214, 216, 218, 220, 222 is constructed of a number of polygons. Individual texture images 208 stored in the texture rendering area 34a are mapped onto the respective polygons, and moved on the objects 214, 216, 218, 220, 222 with time in an arbitrary direction and restored in the texture rendering area 34a.

As a result, it is possible to display a realistic image of water pouring from the faucet 210 and stored in the transparent water tank 212.

[0043]

An example of software (i.e., a rendering means 300) to perform the above function will be described below with reference to FIGS. 8 through 18.

[0044]

The rendering means 300 is supplied to the entertainment apparatus 10 from a randomly accessible recording medium such as a CD-ROM or a memory card or via a network. It is assumed here that the rendering means 300 is read into the entertainment apparatus 10 from an optical disk 42 such as a CD-ROM.

[0045]

The rendering means 300 is downloaded in advance from the optical disk 42 played back by the entertainment apparatus 10 into the main memory 14 of the entertainment apparatus 10 according to a predetermined process, and executed by the main CPU 12.

[0046]

As shown in FIG. 8, the rendering means 300 comprises a table reading means 302 for reading various tables including an object information table 322, a plurality of texture information tables 304, a plurality of polygon information tables 308, and a plurality of movement information tables 312 which are recorded in the optical disk 42, a texture

rendering means 306 for rendering and storing necessary texture images 208 in the texture rendering area 34a of the image memory 34 based on the content of one of the texture information tables 304, an image rendering means 310 for rendering and storing polygons 206 in the display rendering area 34b of the image memory 34 based on the texture images 208 stored in the texture rendering area 34a and the content of one of the polygon information tables 308, and mapping necessary texture images 208 onto the polygons 206, a texture moving means 314 for moving the texture images 208 stored in the texture rendering area 34a based on the content of one of the movement information tables 312 and restoring the moved texture images 208 in the texture rendering area 34a, a texture mapping means 316 for mapping moved texture images 208 onto the polygons 206 stored in the display rendering area 34b, an end determining means 318 for determining whether the processing sequence of the rendering means 300 is completed or not, and a display processing means 320 for outputting image data stored in the display rendering area 34b to the display monitor 16 and displaying an image on the display monitor 16 based on the image data.

[0047]

The various tables used by the rendering means 300 will be described below with reference to FIGS. 9 through 12.

[0048]

As shown in FIG. 9, the object information table 322 stores a plurality of records which correspond to respective

objects, each containing the number of polygons 206 making up an object 204, the address of a corresponding texture information table 304, the address of a corresponding polygon information table 308, and the address of a corresponding movement information table 312.

[0049]

As shown in FIG. 10, the texture information table 304 stores a plurality of records which correspond to respective textures, each containing corresponding record indication information in a corresponding texture table, which stores leading addresses of stored texture image data in respective records, data length of a texture image to be used, and a rendering range (coordinates) for storing the texture image in the texture rendering area 34a.

[0050]

As shown in FIG. 11, the polygon information table 308 stores a plurality of records which correspond to respective polygons, each containing the vertex coordinates of a polygon 206, and the rendering range (coordinates) for storing the texture image 208 to be used in the texture rendering area 34a.

[0051]

As shown in FIG. 12, the movement information table 312 stores a plurality of records which correspond to respective texture images to be moved, each containing a range (coordinates) for storing a texture image to be moved in the texture rendering area, direction information as to the

direction (up, down, left, or right) in which to move the texture image, and circulation information as to whether the texture image is to be circulated or not.

[0052]

The processing sequence of the rendering means 300 will be described below with reference to the flowcharts of FIGS. 13 through 18.

[0053]

In step S1 shown in FIG. 13, the rendering means 300 stores an initial value " 0 " into an index register i used to retrieve objects 204, thereby initializing the index register i.

[0054]

In step S2, the table reading means 302 reads an ith record from the object information table 322.

[0055]

In step S3, the end determining means 318 decides whether the content of the read ith record is an EOF (End-Of-File) code indicative of the end of the object information table 322 or not.

[0056]

If the content of the read ith record is not an EOF code, then control goes to step S4 in which the table reading means 302 reads the address of the corresponding texture information table 304 and the address of the corresponding polygon information table 308 from the ith record in the object information table 322, and reads the

corresponding texture information table 304 and the corresponding polygon information table 308.

[0057]

In step S5, the texture rendering means 306 performs its own processing sequence, which is illustrated in FIG. 15. In step S101 shown in FIG. 15, the texture rendering means 306 stores an initial value " 0 " into an index register j used to retrieve texture images 208, thereby initializing the index register j.

[0058]

In step S102, the texture rendering means 306 reads a jth record from the corresponding texture information table 304.

[0059]

In step S103, the texture rendering means 306 decides whether the content of the read jth record is an EOF code indicative of the end of the texture information table 304 or not.

[0060]

If the content of the read jth record is not an EOF code, then control goes to step S104 in which the texture rendering means 306 reads texture image data based on the record indication information in the texture table and the data length of a texture image 208 stored in the jth record, and stores the texture image data in the portion of the texture rendering area 34a which corresponds to the rendering range that is stored in the jth record.

[0061]

In step S105, the texture rendering means 306 increments the value of the index register j by " 1 ". Then, control goes back to step S102 to repeat the above process based on the content of a next record in the texture information table 304.

[0062]

If the content of the read jth record is an EOF code indicative of the end of the texture information table 304, then the processing sequence of the texture rendering means 306 is ended.

[0063]

Control returns to the main routine shown in FIG. 13. In step S6, the image rendering means 310 performs its own processing sequence, which is illustrated in FIG. 16. In step S201 shown in FIG. 16, the image rendering means 310 stores an initial value " 0 " into an index register k used to retrieve polygons 206, thereby initializing the index register k.

[0064]

In step S202, the image rendering means 310 reads a kth record from the corresponding polygon information table 308.

[0065]

In step S203, a kth polygon 206 is stored into the portion of the display rendering area 34b which corresponds to the rendering range derived from the vertex coordinates.

[0066]

In step S204, the image rendering means 310 reads a texture image 208 from the rendering range for the texture image 208 to be used, among the texture images 208 stored in the texture rendering area 34a, and maps the read texture image 208 onto the kth polygon 206.

[0067]

In step S205, the image rendering means 310 increments the value of the index register k by " 1 ". Thereafter, in step S206, the image rendering means 310 decides whether all the polygons 206 have been processed or not, based on whether or not the value of the index register k is equal to or greater than the number M of polygons.

[0068]

If not all the polygons 206 have been processed, then control returns to step S202 for rendering a next polygon 206 and mapping a necessary texture image 208 onto the next polygon 206.

[0069]

If all the polygons 206 have been processed in step S206, then the processing sequence of the image rendering means 310 comes to an end.

[0070]

Control returns to the main routine shown in FIG. 13. In step S7, the rendering means 300 increments the value of the index register i by " 1 ". Then, control goes back to step S2 for repeating the above process with respect to a next object 204.

[0071]

If the content of the read ith record is an EOF code indicating the end of the object information table 322 in step S3, then control jumps to step S8 in which the display processing means 320 outputs the image data stored in the display rendering area 34b to the display monitor 16, which displays an image based on the supplied image data.

[0072]

In step S9 shown in FIG. 14, the rendering means 300 stores an initial value " 0 " into the index register i used to retrieve objects 204, thereby initializing the index register i.

[0073]

In step S10, the table reading means 302 reads an ith record from the object information table 322.

[0074]

In step S11, the end determining means 318 decides whether the content of the read ith record is an EOF code indicative of the end of the object information table 322 or not.

[0075]

If the content of the read ith record is not an EOF code, then control goes to step S12 in which the table reading means 302 reads the address of the corresponding polygon information table 308 and the address of the corresponding movement information table 312 from the ith record in the object information table 322, and reads the

corresponding polygon information table 308 and the corresponding movement information table 312.

[0076]

In step S13, the texture moving means 314 performs its own processing sequence, which is illustrated in FIG 17. In step S301 shown in FIG. 17, the texture moving means 314 stores an initial value " 0 " into an index register m used to retrieve movement information, thereby initializing the index register m.

[0077]

In step S302, the texture moving means 314 reads an mth record from the movement information table 312.

[0078]

In step S303, the texture moving means 314 decides whether the content of the read mth record is an EOF code indicative of the end of the movement information table 312 or not.

[0079]

If the content of the read mth record is not an EOF code, then control goes to step S304 in which the texture moving means 314 moves a texture image 208 to be moved which corresponds to the rendering range for the texture image 208 stored in the mth record, among the texture images 208 stored in the texture rendering area 34a, by a predetermined interval in the direction information stored in the mth record, and restores the moved texture image 208 in the texture rendering area 34a.

[0080]

In step S305, the texture moving means 314 decides whether the texture image 208 needs to be circulated or not, based on the circulation information stored in the mth record.

[0081]

If the texture image needs to be circulated, then control goes to step S306 in which any texture image 208 pushed out due to the movement carried out in step S304, is stored back in the beginning of the texture rendering area.

[0082]

After the processing in step S306 or if the texture image 208 does not need to be circulated in step S305, then control goes to step S307 in which the texture moving means 314 increments the value of the index register m by " 1 . Thereafter, control goes back to step S302 for moving and restoring the corresponding texture image 208 based on next movement information.

[0083]

If the content of the read mth record is an EOF code indicative of the end of the movement information table 312 in step S303, then the processing sequence of the texture moving means 314 is ended.

[0084]

Control returns to the main routine shown in FIG. 14. In step S14, the texture mapping means 316 performs its own processing sequence.

[0085]

In step S401 shown in FIG. 18, the texture mapping means 316 stores an initial value " 0 " into the index register k used to retrieve polygons 206, thereby initializing the index register k.

[0086]

In step S402, the texture mapping means 316 reads a kth record from the corresponding polygon information table 308.

[0087]

In step S403, the texture mapping means 316 reads a texture image 208 from the rendering range for the texture image 208 to be used, among the texture images 208 stored in the texture rendering area 34a, and maps the read texture image 208 onto the kth polygon 206.

[0088]

In step S404, the texture mapping means 316 increments the value of the index register k by " 1 ". Thereafter, in step S405, the texture mapping means 316 decides whether all the polygons 206 have been processed or not, based on whether or not the value of the index register k is equal to or greater than the number M of polygons.

[0089]

If not all the polygons 206 have been processed, then control returns to step S402 for rendering a next polygon 206 and mapping a necessary texture image 208 onto the next polygon 206.

[0090]

If all the polygons 206 have been processed in step S405, then the processing sequence of the texture mapping means 316 comes to an end.

[0091]

Control returns to the main routine shown in FIG. 14. In step S15, the rendering means 300 increments the value of the index register i by " 1 ". Then, control goes back to step S10 for repeating the above process (i.e., the process of moving the texture image 208 and mapping the texture image 208 onto the polygon 206) with respect to a next object 204.

[0092]

If the content of the read ith record is an EOF code indicating the end of the object information table 322 in step S11, then control jumps to step S16 in which the display processing means 320 outputs the image data stored in the display rendering area 34b to the display monitor 16, which displays an image based on the supplied image data.

[0093]

In step S17, the end determining means 318 decides whether there is a program end request to finish the processing sequence of the rendering means 300 or not. If there is no program end request, then control goes back to step S9 for performing the above process (i.e., the process of moving the texture image 208 and mapping the texture image 208 onto the polygon 206) from the first object 204.

[0094]

If there is a program end request in step S17, then the processing sequence of the rendering means 300 is finished.

[0095]

As described above, the rendering means 300 according to the present invention renders an image by moving a texture image 208 simulatively on at least one object 204. Specifically, one or more semitransparent texture images 208 are placed on the surface of one or more semitransparent or transparent objects 204, and moved simulatively in an arbitrary direction, thereby rendering an image.

[0096]

The texture images 208 that are moved simulatively on the surface of the objects 204 are superposed on the objects 204, thereby generating or animating an infinite number of random patterns to express a fluid motion such as a stream of water or a flow of smoke.

[0097]

Since the rendering means 300 renders an image by moving a texture image 208 simulatively on at least one object 204, the period of time required to calculate and render a fluid motion can greatly be reduced, and a high-quality image can be rendered.

[0098]

It should be understood that various changes and modifications may be made to the embodiment a method of and an apparatus for rendering an image and a recording medium without departing from the scope of the invention.

[0099]

[EFFECT OF THE INVENTION]

A method of and an apparatus for rendering an image and a recording medium according to the present invention renders an image by moving a texture image simulatively on at least one object, therefore, the period of time required to calculate and render a fluid motion can greatly be reduced, and a high-quality image can be rendered.

[BRIEF DESCRIPTION OF THE DRAWINGS]

[FIG. 1]

FIG. 1 is a block diagram of an entertainment apparatus according to an embodiment of the present invention.

[FIG. 2]

FIG. 2 is a perspective view showing the manner in which the water of a river hits a post and flows in divided streams on both sides of the post.

[FIG. 3]

FIG. 3 is a diagram showing the object illustrated in FIG. 2 as it is divided into a number of polygons.

[FIG. 4]

FIGS. 4A and 4B are views of texture images.

[FIG. 5]

FIG. 5 is a diagram illustrative of the directions in which texture images stored in a texture rendering area move.

[FIG. 6]

FIG. 6 is a view showing objects that are arranged in

superposed layers.

[FIG. 7]

FIG. 7A is a view showing the manner in which water pouring from a faucet is stored in a transparent water tank, and FIG. 7B is a view showing the image illustrated in FIG. 7A as it is separated into a plurality of objects.

[FIG. 8]

FIG. 8 is a functional block diagram of a rendering means according to the embodiment of the present invention.

[FIG. 9]

FIG. 9 is a diagram showing details of an object information table.

[FIG. 10]

FIG. 10 is a diagram showing details of a texture information table.

[FIG. 11]

FIG. 11 is a diagram showing details of a polygon information table.

[FIG. 12]

FIG. 12 is a diagram showing details of a movement information table.

[FIG. 13]

FIG. 13 is a flowchart of a processing sequence of the rendering means.

[FIG. 14]

FIG. 14 is another flowchart of a processing sequence of the rendering means.

[FIG. 15]

FIG. 15 is a flowchart of a processing sequence of a texture rendering means.

[FIG. 16]

FIG. 16 is a flowchart of a processing sequence of a rendering means.

[FIG. 17]

FIG. 17 is a flowchart of a processing sequence of a texture moving means.

[FIG. 18]

FIG. 18 is a flowchart of a processing sequence of a texture mapping means.

[DESCRIPTION OF REFERENCE NUMERALS]

10: entertainment apparatus, 16: display monitor, 34: image memory, 34a: texture rendering area, 34b: display rendering area, 42: optical disk, 204: object, 206: polygon, 208: texture image, 214, 216, 218, 220, 222: object, 300: rendering means, 302: table reading means, 306: texture rendering means, 310: rendering means, 314: texture moving means, 316: texture mapping means, 320: display processing means

FIG. 1

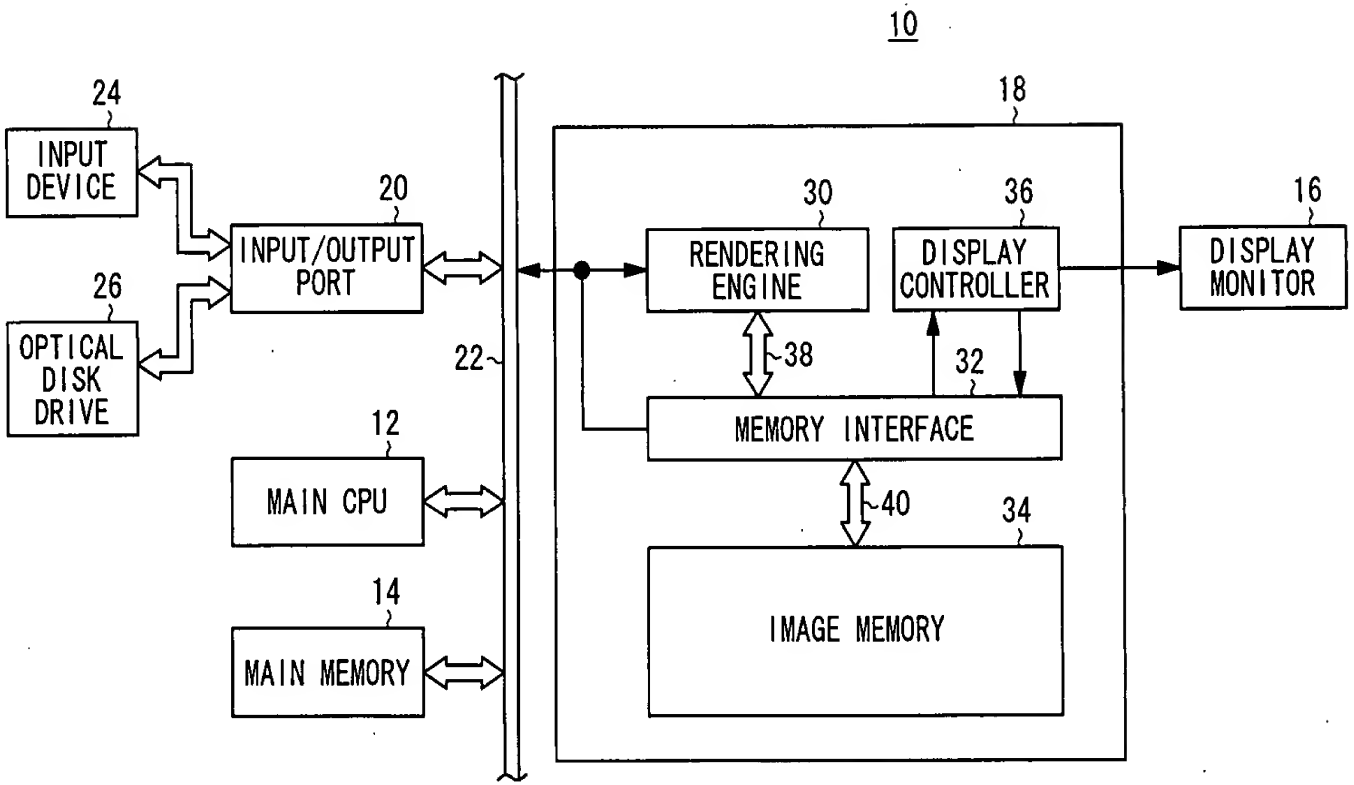

10
24
INPUT DEVICE
26
OPTICAL DISK DRIVE
20
INPUT/OUTPUT PORT
22
12
MAIN CPU
14
MAIN MEMORY
18
30
RENDERING ENGINE
36
DISPLAY CONTROLLER
16
DISPLAY MONITOR
38
32
MEMORY INTERFACE
40
34
IMAGE MEMORY

FIG. 2

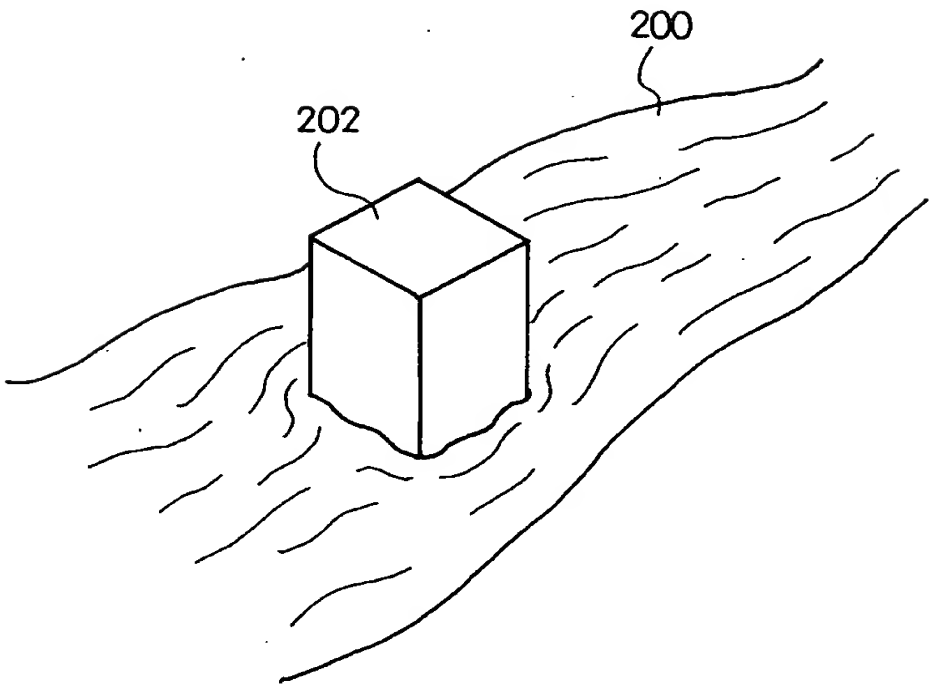


FIG. 3

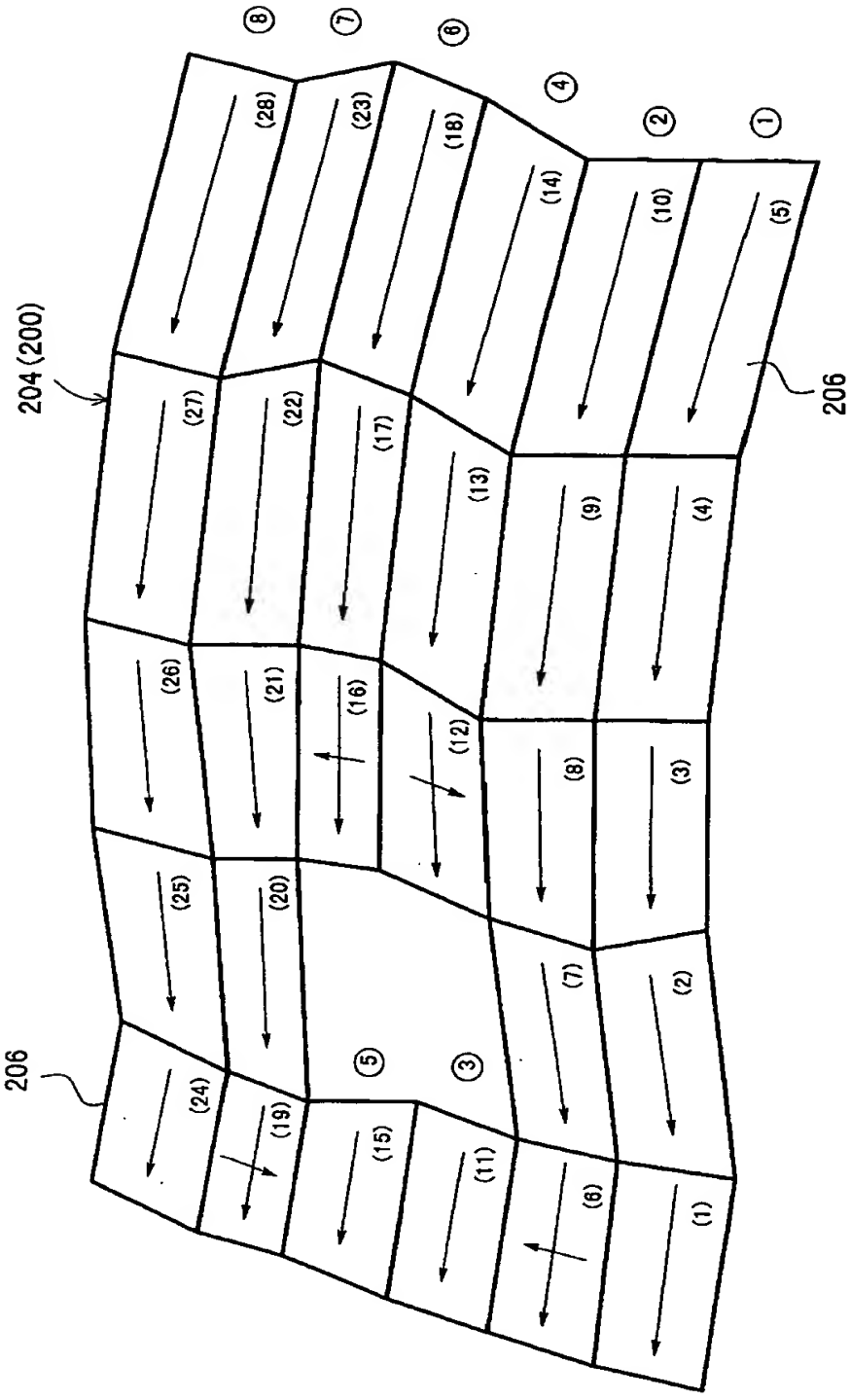

204 (200)
206
206
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)
(27)
(28)

FIG. 4A

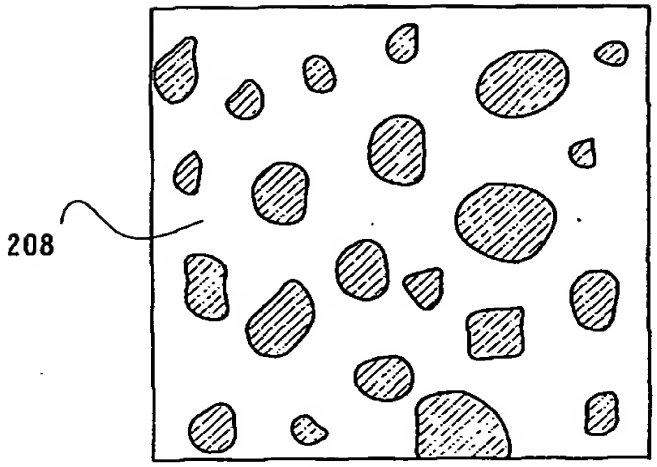


FIG. 4B

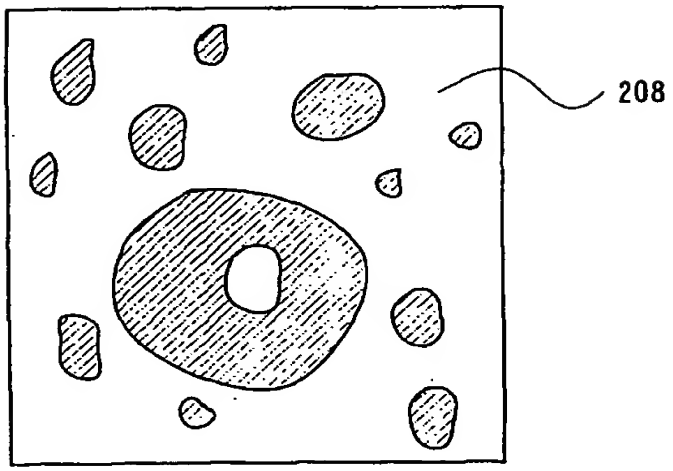


FIG. 5

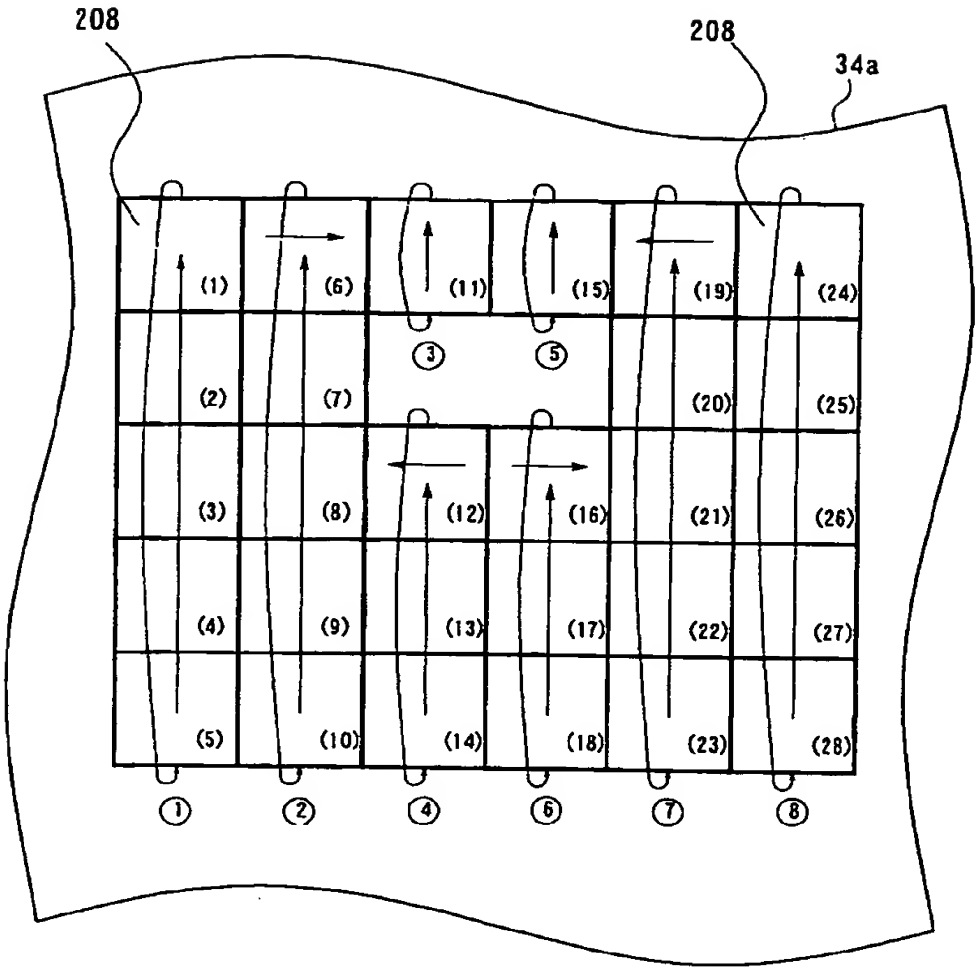

208
208
34a
(1)
(6)
(11)
(15)
(19)
(24)
(2)
(7)
(20)
(25)
(3)
(8)
(12)
(16)
(21)
(26)
(4)
(9)
(13)
(17)
(22)
(27)
(5)
(10)
(14)
(18)
(23)
(28)
①
②
③
④
⑤
⑥
⑦
⑧

FIG. 6

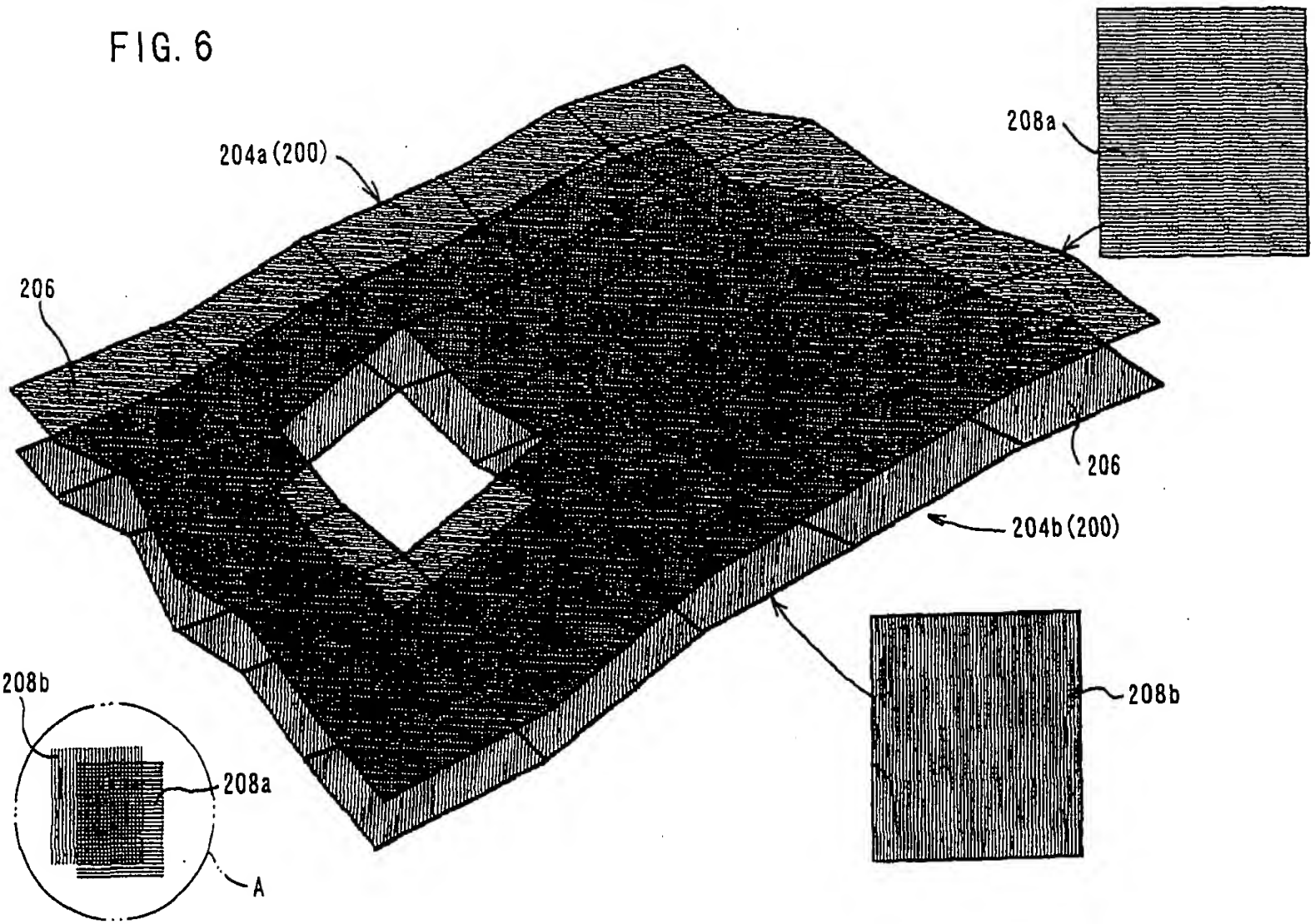


FIG. 7A

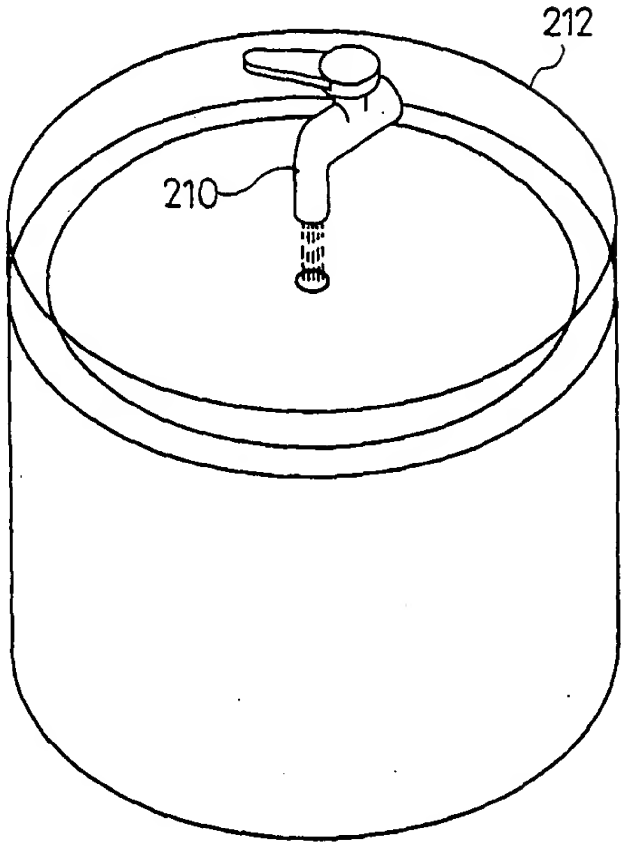


FIG. 7B

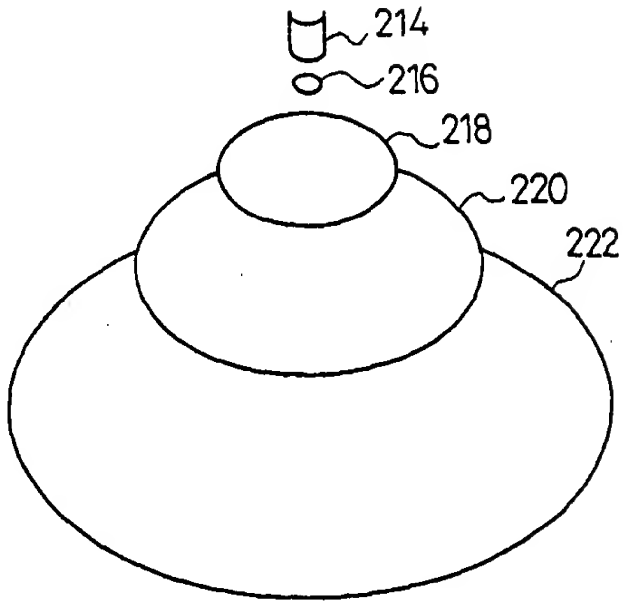


FIG. 8

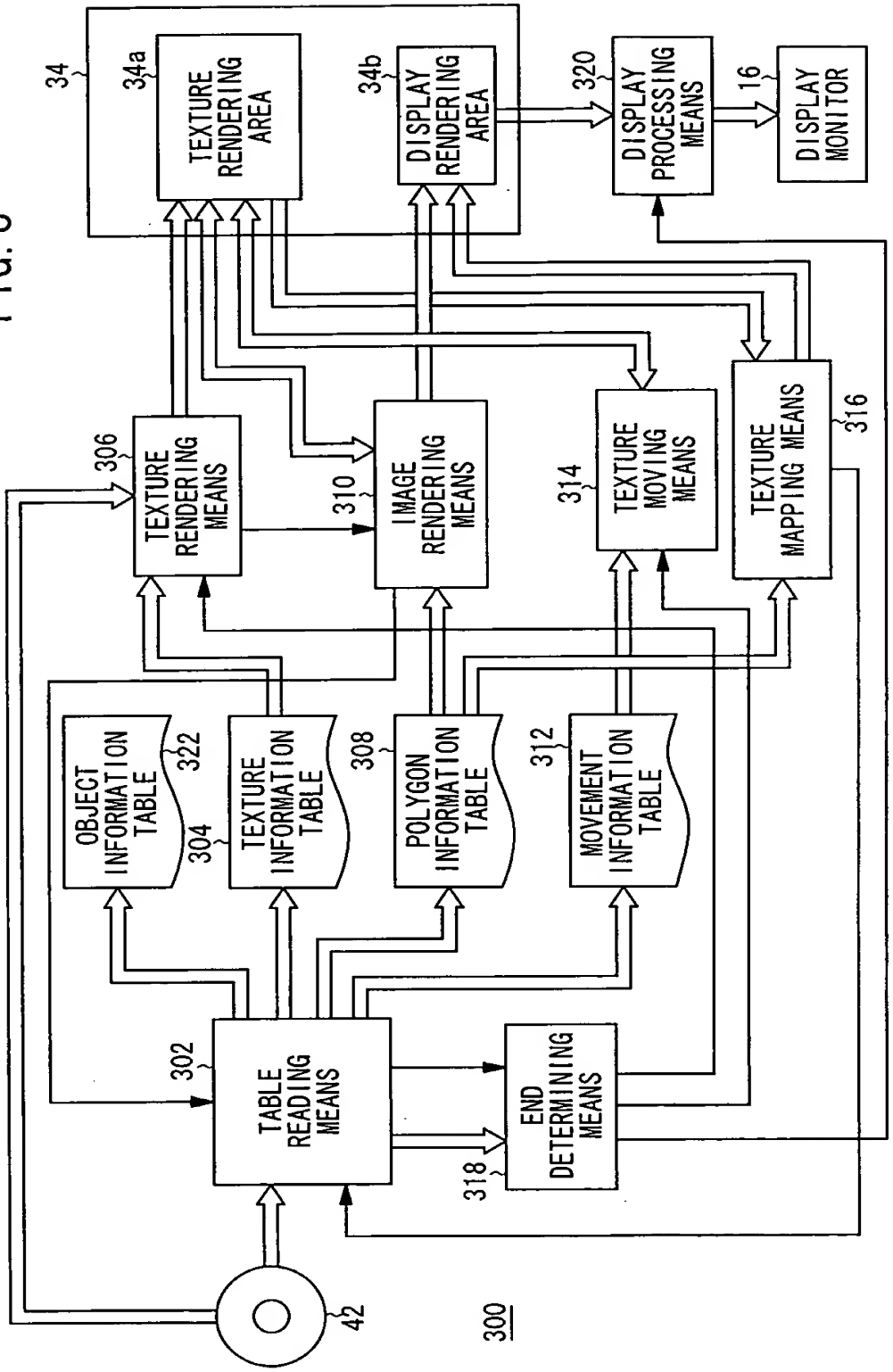

34
34a
TEXTURE RENDERING AREA
34b
DISPLAY RENDERING AREA
320
DISPLAY PROCESSING MEANS
16
DISPLAY MONITOR
306
TEXTURE RENDERING MEANS
310
IMAGE RENDERING MEANS
314
TEXTURE MOVING MEANS
TEXTURE MAPPING MEANS
316
OBJECT INFORMATION TABLE
322
304
TEXTURE INFORMATION TABLE
308
POLYGON INFORMATION TABLE
312
MOVEMENT INFORMATION TABLE
302
TABLE READING MEANS
END DETERMINING MEANS
318
42
300

FIG. 9

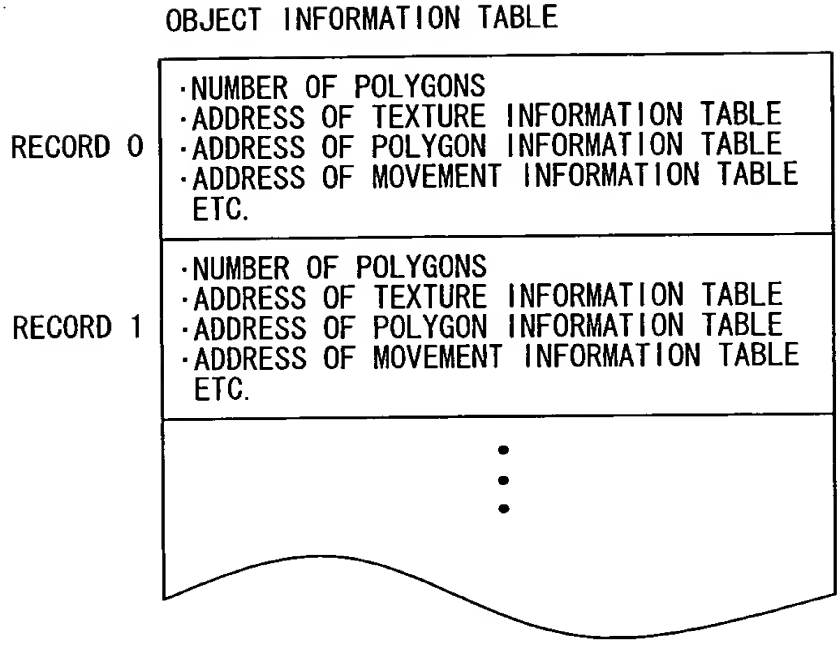

OBJECT INFORMATION TABLE
RECORD 0
·NUMBER OF POLYGONS
·ADDRESS OF TEXTURE INFORMATION TABLE
·ADDRESS OF POLYGON INFORMATION TABLE
·ADDRESS OF MOVEMENT INFORMATION TABLE
ETC.
RECORD 1
·NUMBER OF POLYGONS
·ADDRESS OF TEXTURE INFORMATION TABLE
·ADDRESS OF POLYGON INFORMATION TABLE
·ADDRESS OF MOVEMENT INFORMATION TABLE
ETC.

FIG. 10

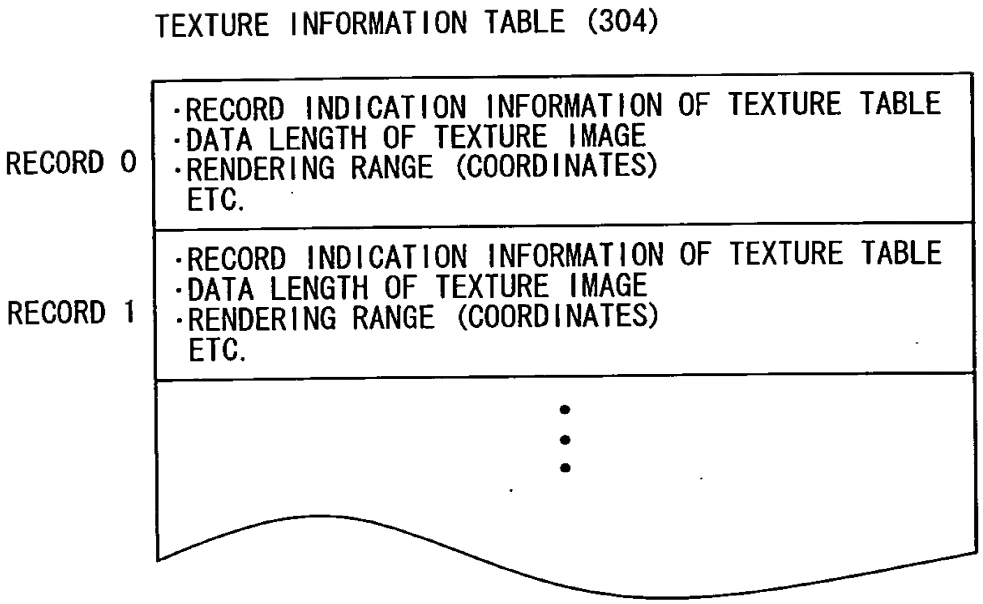

TEXTURE INFORMATION TABLE (304)
RECORD 0
·RECORD INDICATION INFORMATION OF TEXTURE TABLE
·DATA LENGTH OF TEXTURE IMAGE
·RENDERING RANGE (COORDINATES)
ETC.
RECORD 1
·RECORD INDICATION INFORMATION OF TEXTURE TABLE
·DATA LENGTH OF TEXTURE IMAGE
·RENDERING RANGE (COORDINATES)
ETC.

FIG. 11

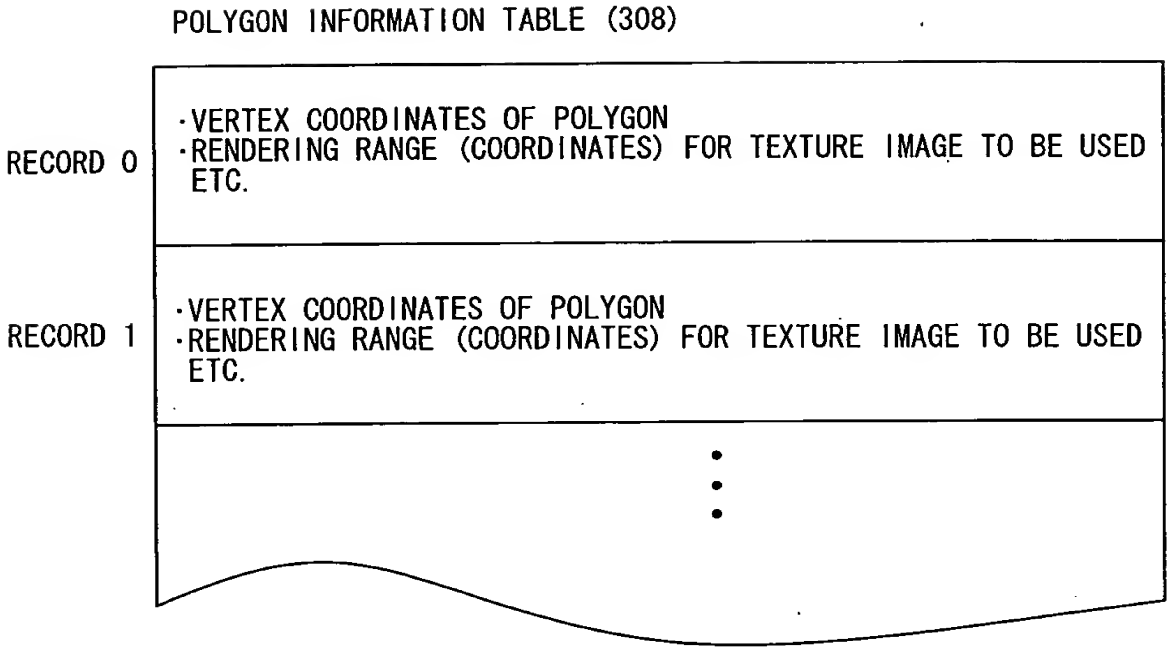

POLYGON INFORMATION TABLE (308)
RECORD 0
·VERTEX COORDINATES OF POLYGON
·RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE USED
ETC.
RECORD 1
·VERTEX COORDINATES OF POLYGON
·RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE USED
ETC.

FIG. 12

MOVEMENT INFORMATION TABLE (312)

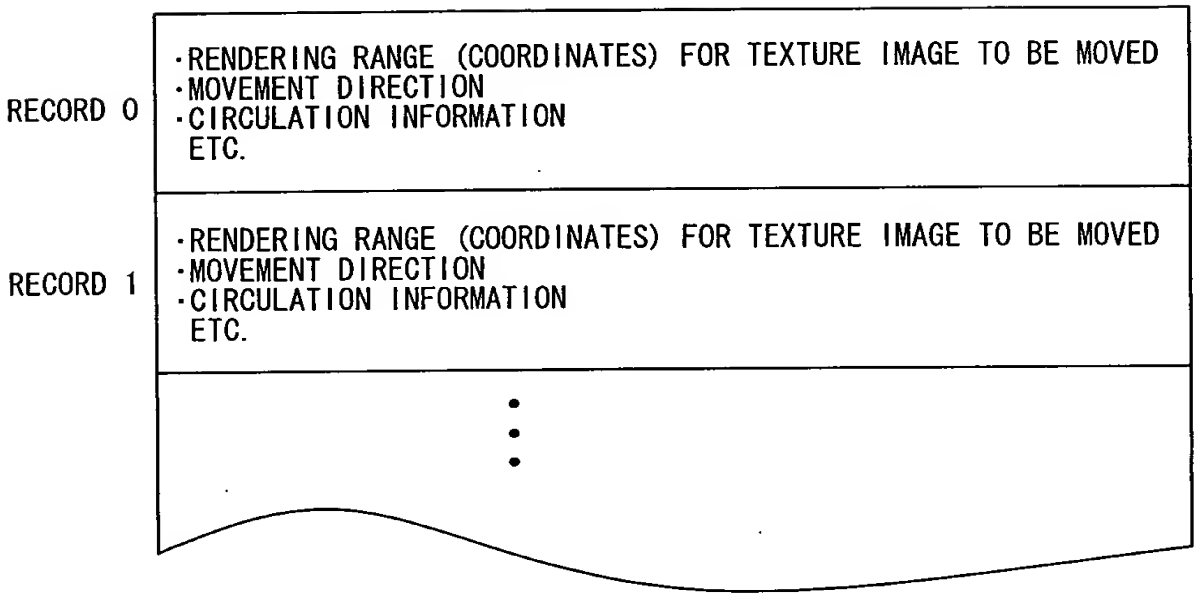


FIG. 13

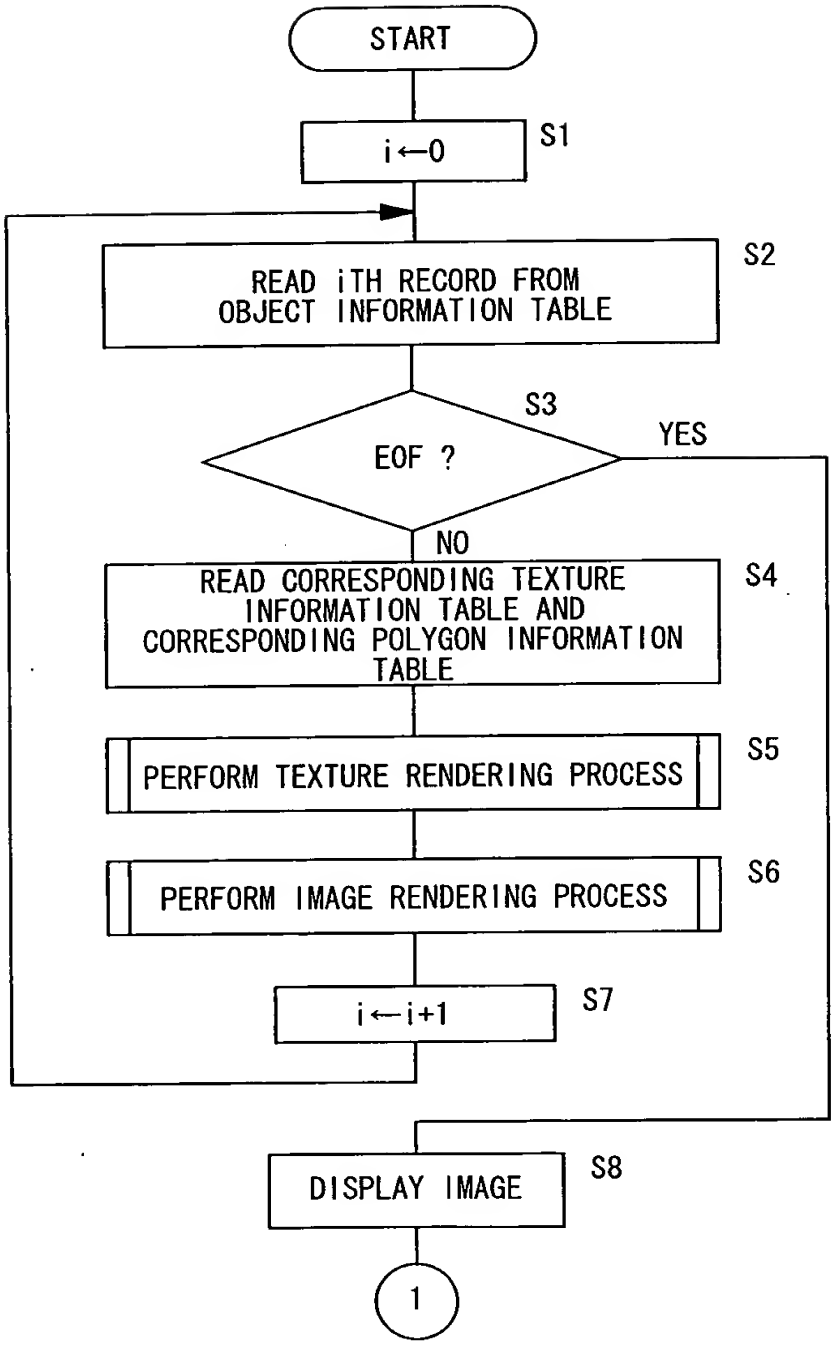

START
i←0
S1
READ iTH RECORD FROM OBJECT INFORMATION TABLE
S2
EOF ?
S3
YES
NO
READ CORRESPONDING TEXTURE INFORMATION TABLE AND CORRESPONDING POLYGON INFORMATION TABLE
S4
PERFORM TEXTURE RENDERING PROCESS
S5
PERFORM IMAGE RENDERING PROCESS
S6
i←i+1
S7
DISPLAY IMAGE
S8
1

FIG. 14

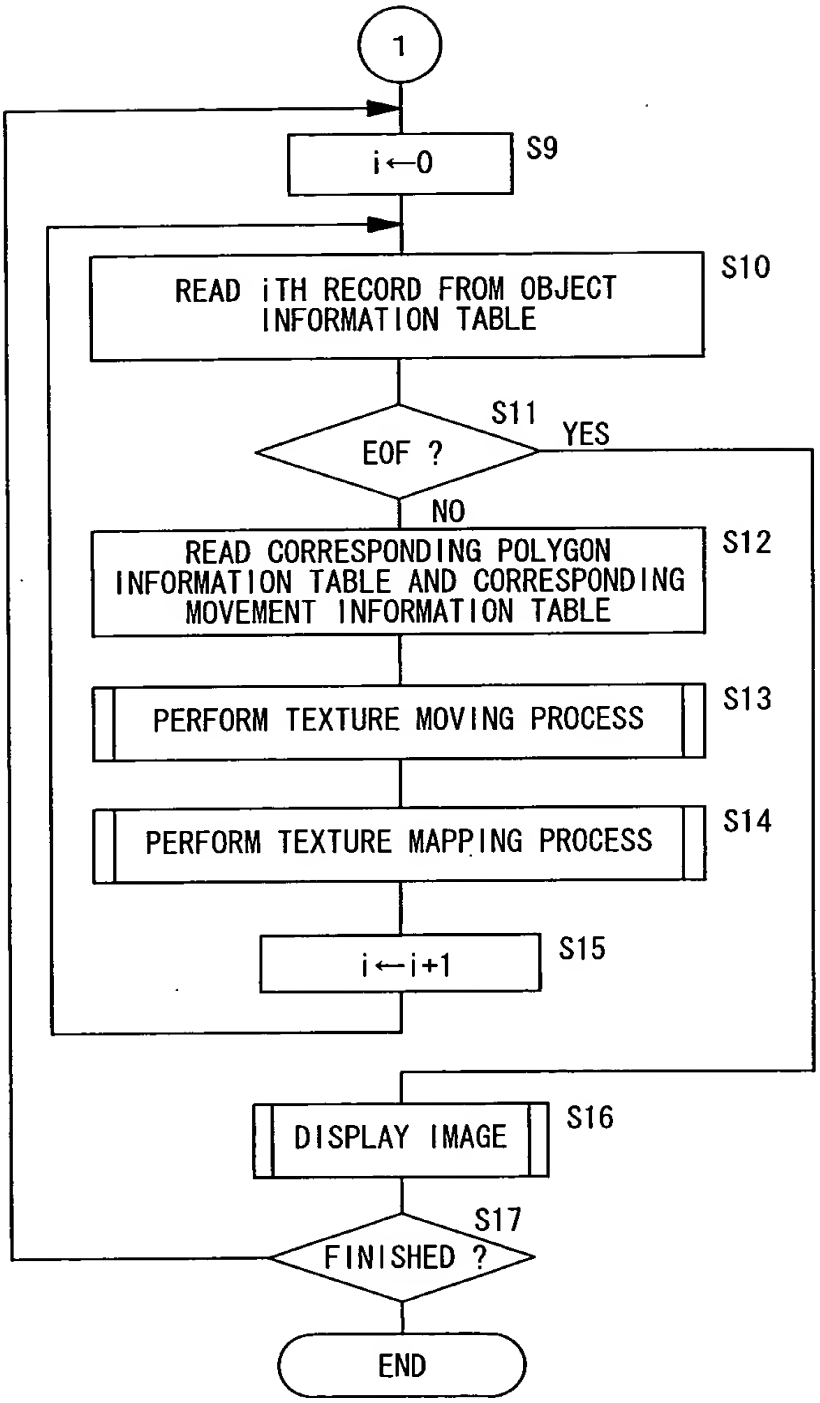

1
i←0
S9
READ iTH RECORD FROM OBJECT INFORMATION TABLE
S10
EOF ?
S11
YES
NO
READ CORRESPONDING POLYGON INFORMATION TABLE AND CORRESPONDING MOVEMENT INFORMATION TABLE
S12
PERFORM TEXTURE MOVING PROCESS
S13
PERFORM TEXTURE MAPPING PROCESS
S14
i←i+1
S15
DISPLAY IMAGE
S16
FINISHED ?
S17
END

FIG. 15

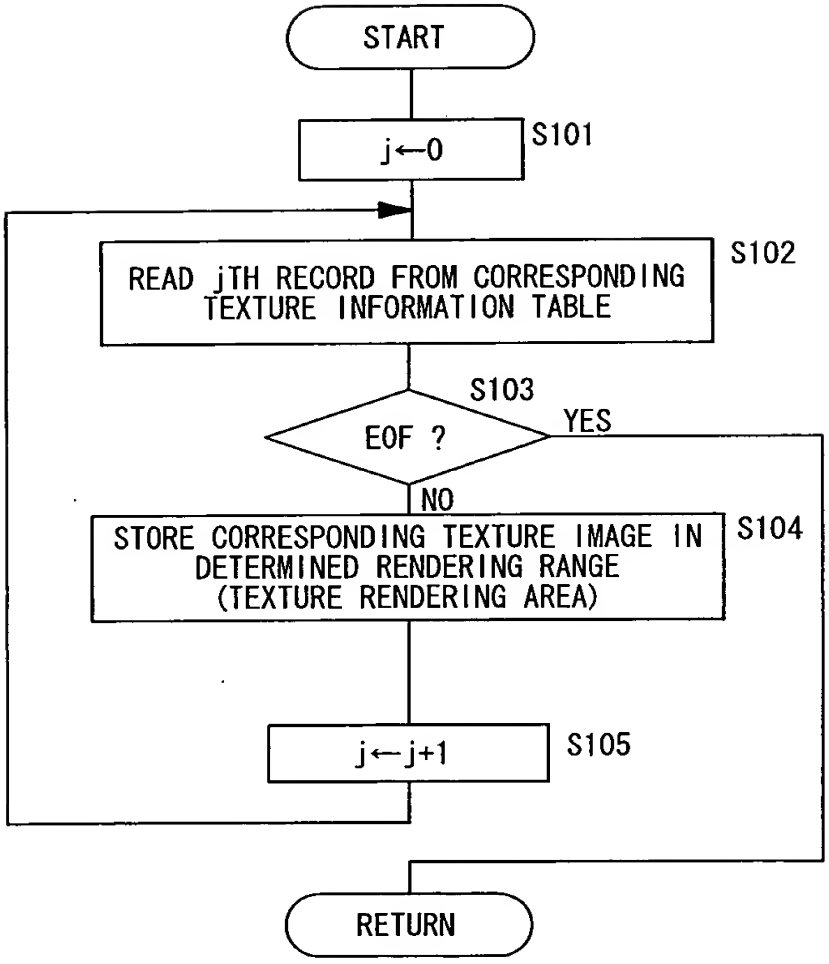

START
j←0
S101
READ jTH RECORD FROM CORRESPONDING TEXTURE INFORMATION TABLE
S102
S103
EOF ?
YES
NO
STORE CORRESPONDING TEXTURE IMAGE IN DETERMINED RENDERING RANGE (TEXTURE RENDERING AREA)
S104
j←j+1
S105
RETURN

FIG. 16

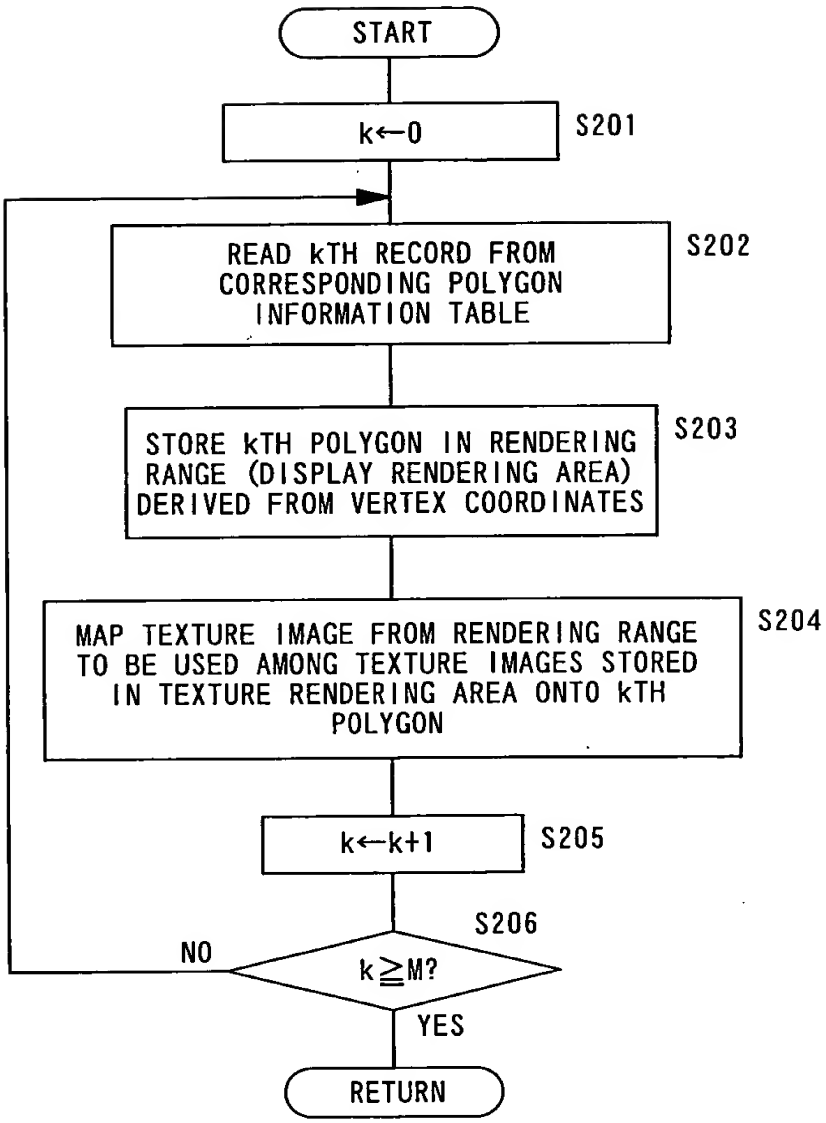

START
k←0
S201
READ kTH RECORD FROM CORRESPONDING POLYGON INFORMATION TABLE
S202
STORE kTH POLYGON IN RENDERING RANGE (DISPLAY RENDERING AREA) DERIVED FROM VERTEX COORDINATES
S203
MAP TEXTURE IMAGE FROM RENDERING RANGE TO BE USED AMONG TEXTURE IMAGES STORED IN TEXTURE RENDERING AREA ONTO kTH POLYGON
S204
k←k+1
S205
S206
NO
k≧M?
YES
RETURN

FIG. 17

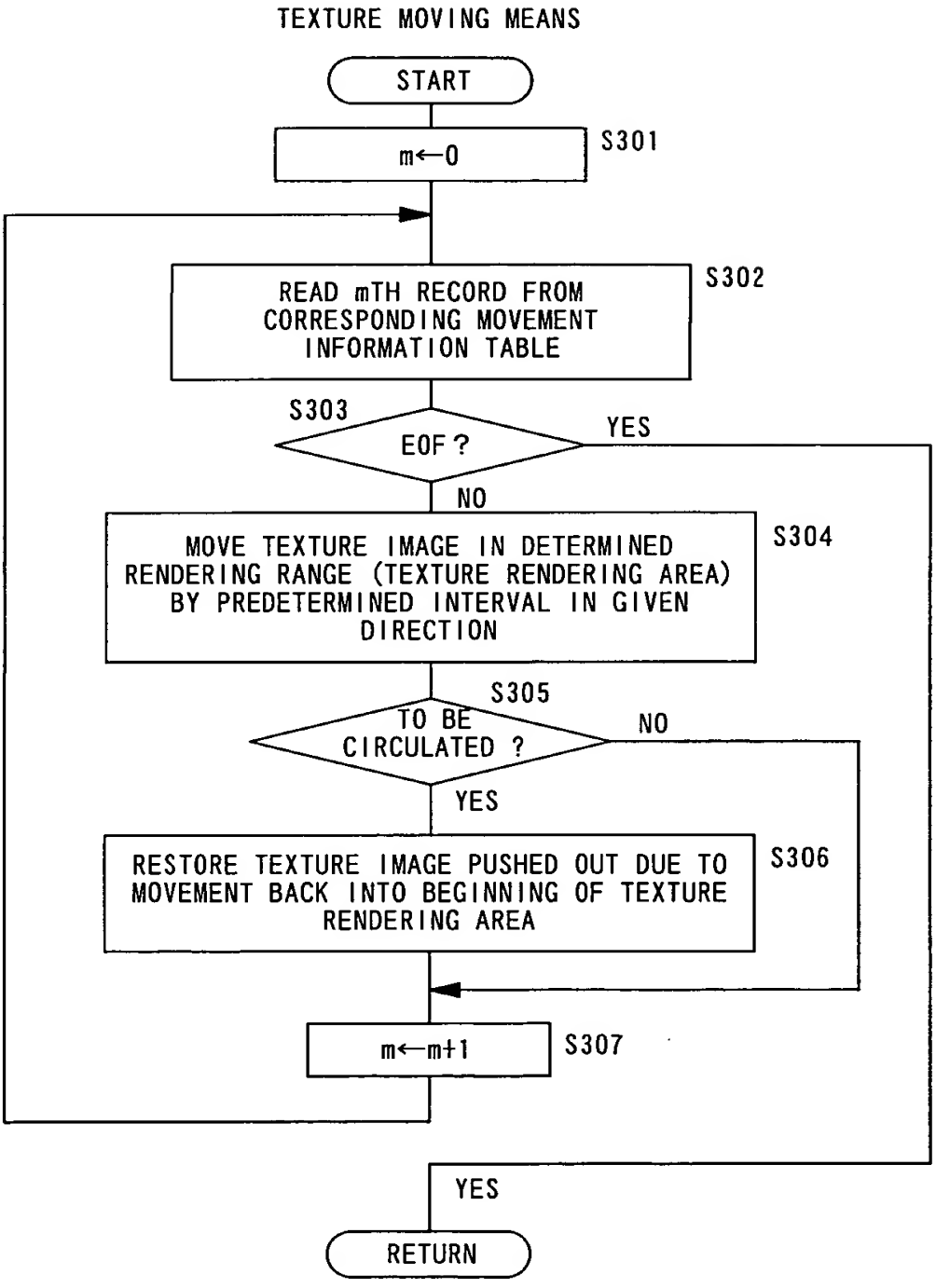
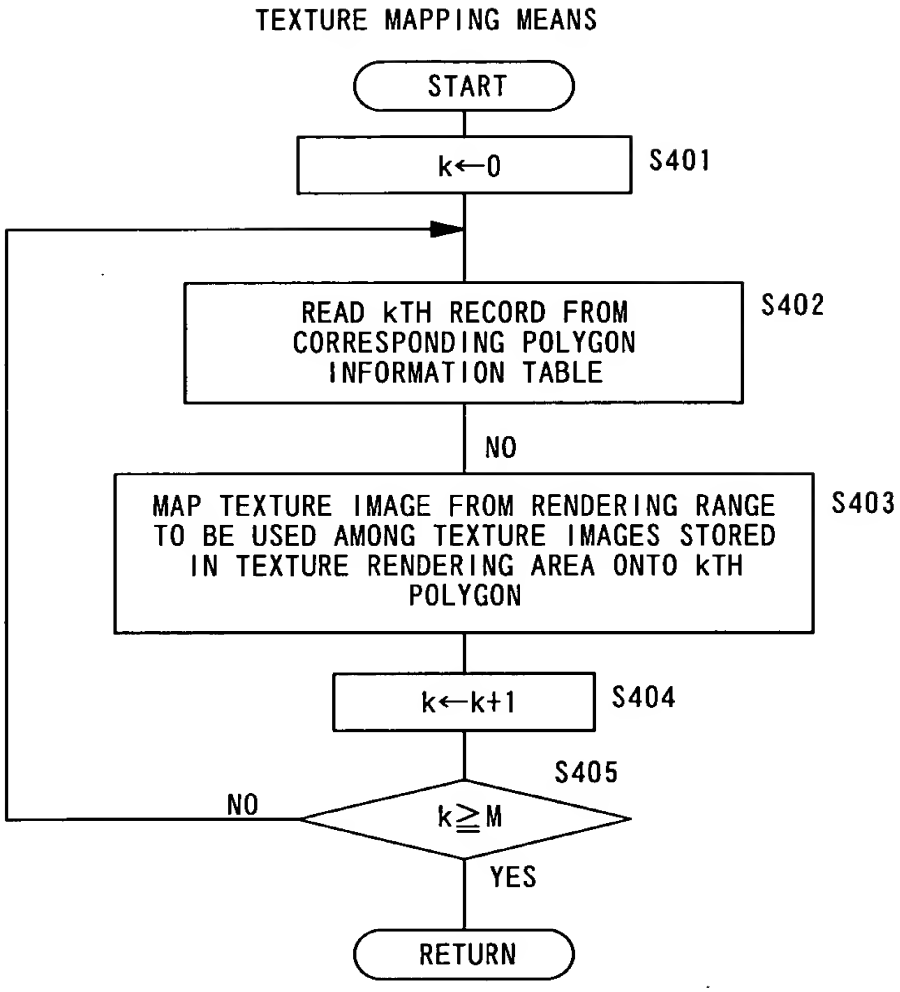

TEXTURE MOVING MEANS
START
m←0
S301
READ mTH RECORD FROM CORRESPONDING MOVEMENT INFORMATION TABLE
S302
S303
EOF ?
YES
NO
MOVE TEXTURE IMAGE IN DETERMINED RENDERING RANGE (TEXTURE RENDERING AREA) BY PREDETERMINED INTERVAL IN GIVEN DIRECTION
S304
S305
TO BE CIRCULATED ?
NO
YES
RESTORE TEXTURE IMAGE PUSHED OUT DUE TO MOVEMENT BACK INTO BEGINNING OF TEXTURE RENDERING AREA
S306
m←m+1
S307
YES
RETURN

FIG. 18


TEXTURE MAPPING MEANS
START
k←0
S401
READ kTH RECORD FROM CORRESPONDING POLYGON INFORMATION TABLE
S402
NO
MAP TEXTURE IMAGE FROM RENDERING RANGE TO BE USED AMONG TEXTURE IMAGES STORED IN TEXTURE RENDERING AREA ONTO kTH POLYGON
S403
k←k+1
S404
S405
NO
k≧M
YES
RETURN

[DOCUMENT NAME] Abstract

[ABSTRACT]

[TASK] To reduce the period of time required to calculate and render a fluid motion such as a stream of water or a flow of smoke, and to render a high-quality image.

[SOLUTION] An apparatus for processing an image has a texture rendering means 306 for storing a necessary texture image 208 based on contents of texture information tables 304 in a texture rendering area 34a of an image memory 34, an image rendering means 310 for storing a polygon 206 in a display rendering area 34b of the image memory 34 based on the texture image 208 stored in the texture rendering area 34a and contents of polygon information tables 308, and mapping the texture image 208 onto necessary polygons 206, a texture moving means 314 for moving the texture image 208 stored in the texture rendering area 34a in an direction based on contents of movement information tables 312 and restoring the moved texture image 208 in the texture rendering area 34a, and a texture mapping means 316 for mapping the moved texture image 208 onto the polygon 206 stored in the display rendering area 34b.

[SELECTED FIGURE] FIG. 8

Applicant's History Information

Identification Number: [395015319]

1. Date of Change	March 31, 1997
[Reason for Change]	Change of address
Address:	1-1, Akasaka 7-chome, Minato-ku, Tokyo
Name:	SONY COMPUTER ENTERTAINMENT INC.